International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

An Improved Data Reduction Technique Based On KNN & NB with Hybrid Selection Method for Effective Software Bugs Triage

Kapil Sahu[1], Dr. Umesh Kumar Lilhore[2], Prof Nitin Agarwal[3]
[1]M. Tech Scholar, Department of CSE, NIIST Bhopal, Madhya Pradesh, India
[2]Head PG, Department of CSE, NIIST Bhopal, Madhya Pradesh, India
[3]Assistant Professor, Department of CSE, NIIST Bhopal, Madhya Pradesh, India

ABSTRACT

In software development process testing process ensures quality management of the product by ensuring bugs free product. Existing methods are based on Naïve byes, SVM methods, which encounters with several issues such as poor precision, recall, TPR and accuracy results. In this research, we are presenting an improved data reduction technique based on kNN & NB with hybrid selection method for effective software bugs triage. Reasons behind the selection of two methods are, k nearest neighbor technique will help in word counts from bug report data and Naïve byes method helps to measure the frequency of the word. The proposed method uses bug report classification, bug report retrieval, and bug report triage. In this proposed method we are also using hybrid selection method for reducing the database, feature selection, and Instance selection methods. Existing method Naïve byes and proposed (kNN + NB with Hybrid selection) are implemented over MATLAB simulator and various performance measuring parameters such as precision, recall, accuracy, detection time and TPR are calculated. An experimental study clearly shows that our proposed method shows outstanding in terms of all the performance measuring parameters as compared to the existing method for bug triage and data reduction.

Keywords- Bug triage, Data Mining, Naïve Byes, kNN, Instance selection, Feature selection

I. INTRODUCTION

A computer code bug is a miscalculation or fault during a program which causes the package to behave in uncaused ways in which. A computer software bugs area unit sometimes annoying and inconvenient for developers, usually leading to serious consequences. Massive software systems come to use bug trailing repositories wherever the users and developers report all the bugs they encounter. The developers attempt to reproduce the bugs with the help of data provided by a newsperson in a bug report then build the desired corrections within the ASCII text file to rectify the difficulty [1].

However, generally, it's impracticable to breed the reportable bug with the data mere in a bug report. In such a state of affairs, the bug is marked with resolution "Non-Reproducible" or "works for me" [3]. This research paper is organized in following chapters, Introduction, data reduction, and Bug triage, existing work, problem statement, proposed solution, result in analysis and finally covers conclusions of the work.

II. DATA REDUCTION & BUG TRIAGE

Data reduction is that the transformation of numerical or alphabetical digital info derived by trial and error or by experimentation into a corrected,

ordered, and simplified kind. The essential construct is that the reduction of innumerable amounts of information all the way down to the significant elements [4]. Bug triage can be defined as "Bug triage may be a method wherever hunter problems square measure screened and prioritized. Triage ought to facilitate guarantee and we have a tendency to suitably manage all rumored problems with bugs yet as enhancements and have requested" [6].

III. EXISTING METHODS

Data mining techniques are widely used in data reduction from large datasets. These methods help in efficient Bug-triage in bug management process. In this research work following research papers has been used for review work.

Paper Reference	Method used	Key Concept
[1]	Feature selection algorithm (FS) and instance selection algorithm (IS), Naïve Byes Method used	➢ Predicting Bug Triage using Data Reduction Methods. ➢ The proposed system performance is verified using Mozilla bug data set.
[2]	Feature selection algorithm (FS) and instance selection algorithm (IS)	➢ Done the testing over 100 bug reports from Eclipse and Mozilla with a change in numbers of assign and non-assign developers.
[3]	Self-Organizing Map, JNM use for Bugs Data Clustering	➢ Improved SOM using Jaccard similarity New Measure produces better results, while it has almost same samples clustered in one group of clusters.
[4]	Classified and detect software bug by J48, ID3 and Naïve Bayes data mining algorithms.	➢ Naïve Bayes 100% correctly classified but with some error and ID3 95% correctly classified, so it is clear that J48 is the best in three respective algorithms so it is more accurate.
[5]	Use Explicit Semantic Analysis (ESA) to carry out the concept-based classification of software defect reports.	➢ Compute the "semantic similarity" between the defect type labels and the defect report.
[6]	Machine Learning Techniques (J48 & AdaBoost) for Classification	➢ The J48 algorithm always gives better results than the AdaBoost algorithm
[8]	Root-Cause and Defect Analysis based on a Fuzzy Data Mining Algorithm	➢ The application of the association rule algorithm developed in this paper is illustrated based on a net making process at

		a netting plant.
[9]	Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques	➢ Review different categories of works in this domain, discuss both advantages and shortcomings and point out challenges and some uncharted territories in the field.
[10]	Combining text mining and data mining for bug report classification	➢ A prototypical recommender system has been developed to demonstrate the applicability of our approach.
[11]	Application of EAs to the DM process is usually named evolutionary data mining (EDM)	➢ Mostly EAs is used in order to enhance the existent ML techniques.
[12]	Spatio-Temporal Data Mining	➢ discuss different types of spatiotemporal data and the relevant data mining questions that arise in the context of analyzing each of these datasets

IV. PROBLEM STATEMENTS

In bug, repositories block the techniques of automatic bug triage. Since software system bug knowledge area unit a sort of free-form text knowledge (generated by developers), it's necessary to come up with well-processed bug knowledge to facilitate the appliance.

Existing method encounters with several issues such as-

- **Precision-** Existing methods have challenges in precision results. Higher precision value for a method shows better results.
- **Recall-** Existing methods have challenges in recall results. Higher recall value shows better results.
- **Accuracy-** Existing methods have challenges in accuracy results. A higher value of accuracy is always desirable for any best algorithm.
- **True positive rate-** Existing methods have challenges in true positive rate results. A higher result for TPR shows better performance for any method.
- **Detection time -** Existing methods have challenges in detection time or data reduction time results. For any method, lesser detection time shows better performance.

4.1 OBJECTIVE

In this research, we are presenting an improved data reduction technique based on kNN + NB with hybrid selection method for effective software bugs triage. The main objective of this research work is to overcome the problems which are described in chapter 4.1.

The main objective of the research work is as follows-

- **Precision-**Higher precision value for a method shows better results. The proposed method will achieve better precision results.
- **Recall-**Higher recall value shows better results. The proposed method will achieve better recall results.
- **Accuracy-**Higher value of accuracy is always desirable for any best algorithm. The proposed method will achieve better accuracy results.
- **True positive rate-** A higher result for TPR shows better performance for any method. The proposed method will achieve better TPR results.

➢ **Detection time-** For any method lesser detection time shows better performance. The proposed method will achieve better detection time results.

V. PROPOSED SOLUTION

Data reduction techniques are widely used for bug triage. It attracts researcher to work in the field of efficient data reduction technique for effective and efficient bug triage. Existing methods are based on Naïve byes, SVM methods, which encounters with several issues such as poor precision, recall, TPR and accuracy results. In this research, we are presenting an improved data reduction technique based on KNN + NB with hybrid selection method for effective software bugs triage [5].

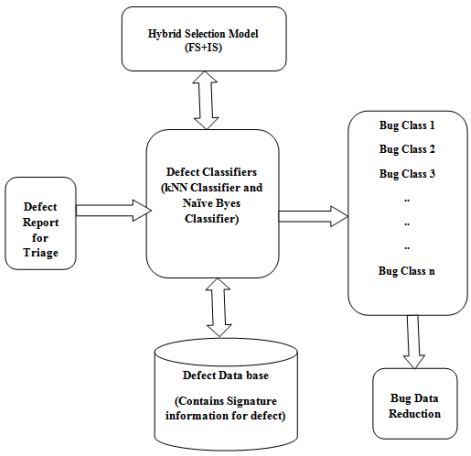

Figure 5.1 Working of Proposed Method

Reasons behind the selection of two methods are, K nearest neighbor technique will help in word counts from bug report data and Naïve byes method helps to measure the frequency of the word. The proposed method uses bug report classification, bug report retrieval, and bug report triage. In this proposed method we are also using hybrid selection method for reducing the database, feature selection, and Instance selection methods [7].

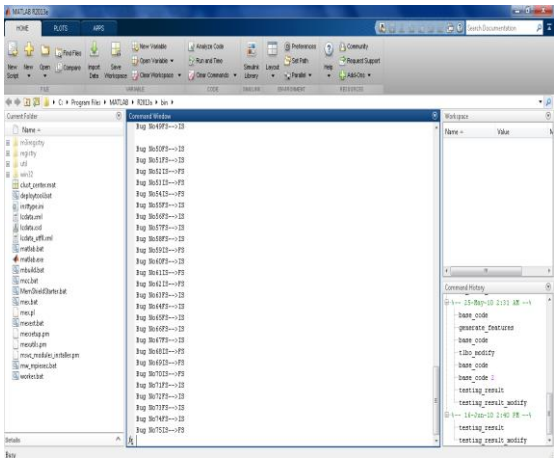

Figure 5.1.2 Simulation of proposed method

5.1 DATA SET-

For simulation of existing method and proposed method following eclipse and Mozilla open source, data set were used.

- Data set link for eclipse https://raw.githubusercontent.com/ansymo/msr2013bug_dataset/master/data/v02/eclipse/assigned_to.json
- Data set link for Mozilla data set- https://github.com/ansymo/msr2013-bug_dataset/tree/master/data/v02/mozilla

VI. RESULT ANALYSIS

In this research, we are presenting an improved data reduction technique based on KNN + NB with hybrid selection method for effective software bugs triage. Existing method Naïve byes and proposed (KNN + NB with Hybrid selection) are implemented over MATLAB simulator and various performance measuring parameters such as precision, recall, accuracy, detection time and TPR are calculated. Simulation results for Mozilla dataset & Eclipse dataset (Bug reports).

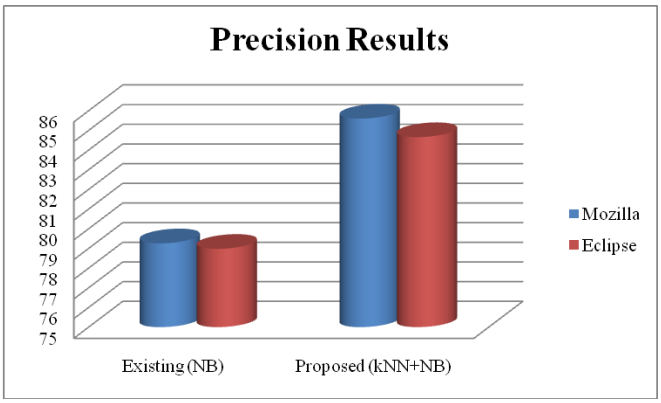

Figure 6.1 Simulation Results for Precision

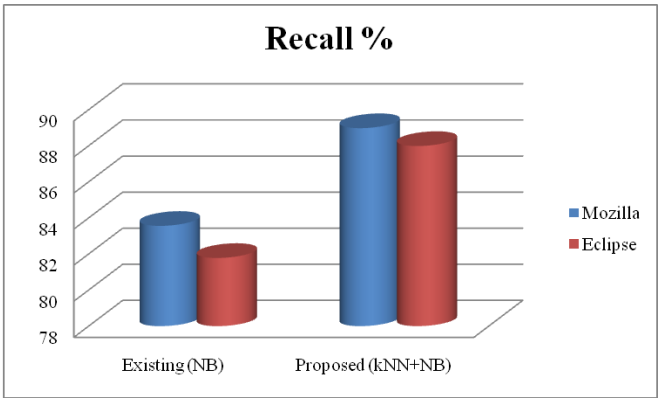

Figure 6.2 Simulation Results for Recall

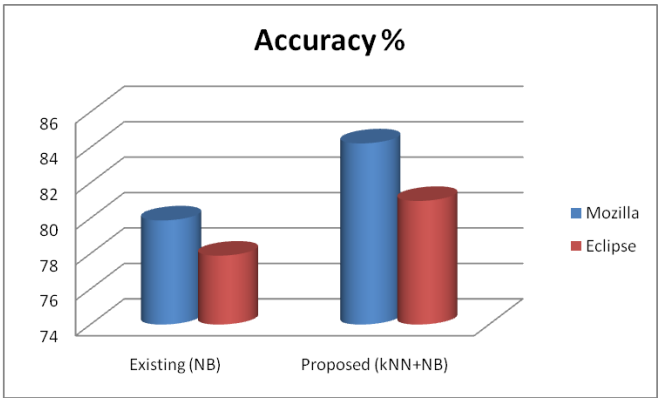

Figure 6.3 Simulation Results for Accuracy

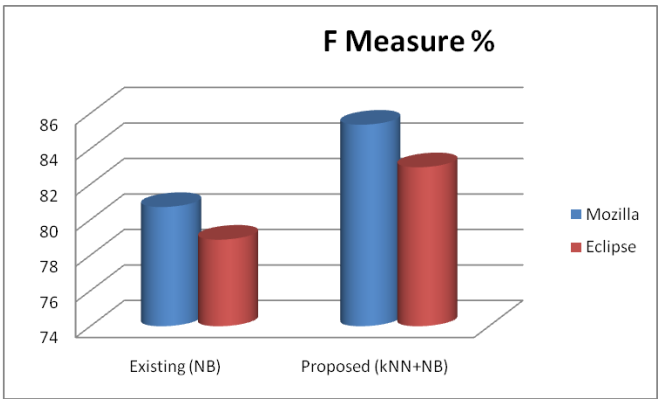

Figure 6.4 Simulation Results for F-measure

The above results clearly show that proposed method shows better results over existing metho.

VII. CONCLUSIONS & FUTURE WORKS

Data reduction techniques are widely used for bug triage. It attracts researcher to work in the field of efficient data reduction technique for effective and efficient bug triage. Existing methods are based on Naïve byes, SVM methods, which encounters with several issues such as poor precision, recall, TPR and accuracy results. In this research, we have presented an improved data reduction technique based on KNN + NB with hybrid selection method for effective software bugs triage. The proposed method and existing method (NB) is implemented over MATLAB simulator; an experimental study clearly shows that our proposed method shows better results over the existing method in terms of better precision, recall, TPR, and accuracy.

In this research, we have presented an improved data reduction technique based on KNN + NB with hybrid selection method for effective software bugs triage. In future work, we can implement our proposed method with real-time data and use more classifiers for better results.

VIII. REFERENCES

[1]. ShanthiPriya Duraisamy, Laxmi Raja, KalaiSelvi Kandaswamy,"An Approach for Predicting Bug Triage using Data Reduction Methods", International Journal of Computer Applications (0975 – 8887) Volume 177 – No. 5, November 2017, PP 1-6.

[2]. Pooja S. Dhole, Prof. Avinash P. Wadhe, "Anatomization of Bug Triage using Data Reduction Techniques", Satellite Conference ICS SD 2016 International Conference on Science and Technology for Sustainable Development, Kuala Lumpur,Malaysia, May 24-26, 2016, www.internationaljournalssrg.org, PP 124-130.

[3]. Attika Ahmed, Rozaida Ghazali, "An Improved Self-Organizing Map for Bugs Data Clustering",2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 22 October 2016, Shah Alam, Malaysia PP 135-141.

[4]. Dhyan Chandra Yadav, Saurabh Pal,"Software Bug Detection using Data Mining", International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 15, April 2015, PP 21-27.

[5]. Sangameshwar Patil,"Concept-based Classification of Software Defect Reports",2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR),978-1-5386-1544-7/17 $31.00 © 2017 IEEE, PP 182-187.

[6]. Poonam Pandey, Radhika Prabhakar, "An Analysis of Machine Learning Techniques (J48 & AdaBoost) -for Classification", IEEE 2016 Conference CNC-16, PP 978-984.

[7]. Haidar Osman, Mohammad Ghafari,"An Extensive Analysis of Efficient Bug Prediction Configurations", ACM PROMISE Conferences November 8, 2017, Toronto, Canada, PP 978-988.

[8]. Seyed Ali Asghar Mostafavi Sabet, Alireza Moniri, Farshad Mohebbi, "Root-Cause and Defect Analysis based on a Fuzzy Data Mining Algorithm", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 9, 2017, PP 21-29.

[9]. SEYED MOHAMMAD GHAFFARIAN and HAMID REZA SHAHRIARI,"Sofware Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey", ACM Computing Surveys, Vol. 50, No. 4, Article 56. Publication date: August 2017, PP 56-92.

[10]. Yu Zhou, Yanxiang Tong, Ruihang Gu and Harald Gall,"Combining text mining and data mining for bug report classification ", JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS J. Softw. Evol. and Proc. 2016; 28:150–176.

[11]. Rafael Alcala, Maria Jose Gacto, Jesus Alcala Fdez, "Evolutionary data mining and applications: A revision on the most cited papers from the last 10 years (2007–2017)", WIREs Data Mining Knowl Discov. 2017, Wiley, PP 1-17.

[12]. GOWTHAM ATLURI, ANUJ KARPATNE, VIPIN KUMAR, "Spatio-Temporal Data Mining: A Survey of Problems and Methods", ACM Computing Surveys, Vol. 1, No. 1, Article. Publication date: November 2017, PP 1-37.

[13]. D. Kavitha, "SURVEY OF DATA MINING TECHNIQUES FOR SOCIAL NETWORKING WEBSITES", IJCSMC, Vol. 6, Issue. 4, April 2017, pg.418 – 426.

[14]. Naresh.E, Vijaya Kumar B.P, Sahana.P.Shankar, "Comparative Analysis of the Various Data Mining Techniques for Defect Prediction using the NASA MDP Datasets for Better Quality of the Software Product", Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 10, Number 7 (2017) pp. 2005-2017

[15]. ZHANG Jie, WANG XiaoYin, HAO Dan, XIE Bing, ZHANG Lu MEI Hong, "A survey on bug-report analysis", SCIENCE CHINA Information Sciences Sand printer-Verlag Berlin Heidelberg 2015, Vol. 58, PP 1-24.

[16]. Anvik J, Hiew L, Murphy G C. Who should fix this bug? In: Proceedings of the International Conference on Software Engineering, Shanghai, 2006. 361–370.

[17]. Hooimeijer P, Weimer W. Modeling bug report quality. In: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, Atlanta, 2007. 34–43.

[18]. Nguyen AT, Nguyen T T, Nguyen T N, et al. Duplicate bug report detection with a combination of information retrieval and topic modeling. In: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, Essen, 2012. 70–79.

[19]. Xie J, Zhou M, Mockus A. Impact of triage: a study of Mozilla and gnome. In: Proceedings of the ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, 2013. 247–250.

[20]. J. W. Park, M. W. Lee, J. Kim, S. W. Hwang, and S. Kim, "Costriage: A cost-aware triage algorithm for bug reporting systems," in Proc. 25th Conf. Artif. Intell. Aug. 2011, pp. 139–144.

[21]. A. Tamrawi, T. Nguyen, J. Al-Kofahi, and T. Nguyen, "Fuzzy set and cache-based approach for bug triaging," in Proc. 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, 2011, pp. 365–375.

[22]. Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis, "Efficient ticket routing by resolution sequence mining," in Proc. 14th ACM SIGKDD

Int. Conf. Knowl. Discovery Data Mining, Aug. 2008, pp. 605–613.

[23]. Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, "Towards Effective Bug Triage with Software Data Reduction Techniques", IEEE transactions on knowledge and data engineering, vol. 27, no. 1, January 2015.

[24]. T. Zhang, G. Yang, B. Lee, I. Shin "Role Analysis-based Automatic Bug Triage Algorithm", 2012.

[25]. P. Bhattacharya, L. Neamtiu, C. R. Shelton "Automated, highly-accurate, bug assignment using Machine learning and tossing graphs", 2012.

[26]. https://raw.githubusercontent.com/ansymo/msr2013bug_dataset/master/data/v02/eclipse/assigned_to.json

[27]. https://github.com/ansymo/msr2013-bug_dataset/tree/master/data/v02/mozilla